**System-level Design**

**Overview**

The MVC architecture splits the application into three primary components:

1. Model: Manages the data, logic, and rules of the application.
2. View: Presentation layer that displays information to the user.
3. Controller: Takes user input from the View, processes it (with possible updates to the Model), and returns the output display to the View.

**Models**

**TA Applicant Model**

* Handles storing applicant data, including CVs.
* Methods for application submission.

**Department Staff Model**

* Manages courses requiring TAs.
* Handles the logic for preliminary TA-to-course matching.

**TA Committee Model**

* Manages decision-making functionalities.
* Stores records of TA assignments.

**Instructor Model**

* Manages the assessment data of TAs.
* Stores feedback for TAs.

**Common Features**

* All models interact with Azure SQL Database for data storage.
* CVs and other documents will be stored using Azure Blob Storage.

**Views**

**TA Applicant View**

* Forms for submitting applications and CVs.
* Displays courses they can apply for.

**Department Staff View**

* Dashboard to manage course entries and view TA applicants.
* Preliminary recommendation interface.

**TA Committee View**

* Interface for reviewing TA applicants and making decisions.
* Overview of TA assignments.

**Instructor View**

* Interface for submitting TA performance evaluations.

**Controllers**

**TA Applicant Controller**

* Handles form submissions for applications.
* Manages authentication via Azure Active Directory.

**Department Staff Controller**

* Manages the addition and modification of course details.
* Handles the logic for generating preliminary recommendations.

**TA Committee Controller**

* Takes input for TA assignments and updates the model accordingly.

**Instructor Controller**

* Takes performance evaluations and updates the model.

**Common Features**

* Azure Logic Apps or Azure Functions can be used to automate routine tasks like generating preliminary recommendations.
* Azure Notification Hubs for notification handling.

**Data Flow**

1. TA Applicants: After authentication via Azure AD, they submit forms. The Controller processes this data, updates the Model, and stores it in Azure SQL Database and Blob Storage.
2. Department Staff: Add or modify course details. The Controller processes these changes and updates the Model accordingly.
3. TA Committee: Reviews TA applicant data and makes decisions. The Controller processes this and updates the Model.
4. Instructors: Submit TA evaluations via a form. The Controller processes these and updates the Model.

**Security & Compliance**

1. Authentication through Azure Active Directory.
2. Role-based access control to define roles like TA applicant, department staff, etc.

**Detailed Design: Decision-making and Approval Subsystem**

**Overview:**

The Decision-making and Approval Subsystem is designed to assist the TA Committee in the review and approval process of TA-to-course assignments. This subsystem includes a specialized dashboard for the TA Committee and uses Azure SQL Database to record and store all decisions regarding TA assignments.

**Components:**

1. TA Committee Dashboard: A web interface that allows TA Committee members to review applications, matching recommendations, and make decisions.
2. Azure SQL Database: Utilized for recording and storing the TA Committee’s decisions.

**Data Models:**

Committee Decision Entity in Azure SQL Database

* Decision ID (Primary Key)
* Committee Review ID (Foreign Key)
* TA ID
* Course ID
* Decision (Approved/Rejected/Deferred)
* Reason (Optional)
* Timestamps (created\_at, updated\_at)

**Functionalities and Interactions:**

Review of Preliminary Recommendations

TA Committee Dashboard

* Displays a list of preliminary TA-to-course recommendations generated by the Matching and Recommendation Subsystem.
* Allows for sorting and filtering based on various criteria like matching score, TA qualifications, etc.

Azure SQL Database

* Fetches the required data based on the TA Committee’s sorting or filtering preferences.

Decision-making

TA Committee Dashboard

* Provides options for each preliminary recommendation: “Approve,” “Reject,” or “Defer.”
* Allows for inputting reasons, especially in the case of “Rejected” or “Deferred” decisions.

Azure SQL Database

* Records the committee’s decisions and reasons, if provided.

Bulk Actions

TA Committee Dashboard

* Allows committee members to approve or reject multiple recommendations in a batch operation.

Azure SQL Database

* Records all batch operations appropriately.

History and Auditing

TA Committee Dashboard

* Provides a history tab to view past decisions.

Azure SQL Database

* Stores historical data for auditing purposes.

**API Endpoints:**

1. /committee/recommendations - GET: Retrieves all preliminary recommendations for the TA Committee.
2. /committee/decisions - POST: Records the TA Committee's decisions.
3. /committee/history - GET: Retrieves the history of committee decisions.

**Security Measures:**

1. Authentication: Only authenticated TA Committee members can access the dashboard.
2. Authorization: Role-based access control ensures only authorized committee members can make or view decisions.
3. Data Validation: All decisions and reasons are validated before recording.
4. Audit Logs: Every decision is logged for auditing purposes.

**Monitoring:**

Azure Monitoring Tools: Utilized for logging, auditing, and real-time monitoring of subsystem activities.

By providing an effective decision-making interface and utilizing Azure SQL Database for secure data storage, the Decision-making and Approval Subsystem aims to simplify the TA assignment approval process for the TA Committee.

**>Component Diagram**

Web Application (Frontend)

* Built using HTML, CSS, JavaScript
* Interacts with Business Logic via REST APIs

Business Logic (Backend)

* Developed in Python
* Handles CRUD operations, matching algorithms, and notifications
* Interacts with the Data Access Layer

Data Access Layer

* Azure SQL Database for structured data
* Azure Blob Storage for unstructured data like CVs
* Responsible for all data storage tasks

User Management

* Azure Active Directory for authentication and RBAC
* Interacts with Business Logic and Data Access Layer

Performance Evaluation

* Processes evaluations submitted by instructors
* Interacts with Business Logic and Data Access Layer

Automated Workflows

* Azure Logic Apps or Azure Functions
* Interacts with Business Logic and Data Access Layer

Monitoring and Maintenance

* Uses Azure Monitoring tools
* Interacts with all layers for logging and monitoring

>Component Diagram

